Journal of Computer Science

Research Article

A Deep Learning Approach for Telugu Domain Identification with Multichannel LSTM-CNN

Buddha Hari Kumar[1], Chitra Perumal[1], Inakoti Ramesh Raja[2], Chukka Ramesh Babu[3], Srinivas Rao Gorre[4] and Santosh Tripurana[5]

[1]*Electronics and Communication Engineering Department, Sathyabama Institute of Science and Technology, Chennai, India*
[2]*Department of Electronics and Communication Engineering, Aditya University, Surampalem, India*
[3]*Department of Electronics and Communication Engineering, Vignan's Institute Of Information Technology, Visakhapatnam, India*
[4]*Department of Electronics and Communication Engineering, Vasavi College of Engineering, Hyderabad, Telangana, India*
[5]*Department of Electronics and Communication Engineering , Vignan's Institute of Engineering for Women, Visakhapatnam, India*

Article history
Received: 30-01-2025
Revised: 06-03-2025
Accepted: 24-03-2025

Corresponding Author:
Buddha Hari Kumar
Electronics and Communication Engineering Department,
Sathyabama Institute of Science and Technology, India
Email:
harikumarbuddha67@gmail.com

Abstract: The vast growth of textual data has ushered into the limelight, a plethora of applications in information retrieval and natural language processing (NLP). Proper extraction of information from text is heavily dependent on recognizing the thematic content, which becomes crucial in the tasks of document summarization, information extraction, question answering, machine translation, and sentiment analysis. The great complexity of this challenge arises for regional languages such as Telugu, where unique linguistic features demand specialized approaches. In this work, we propose a Telugu Technical Domain Identification model based on a Multichannel Long Short-Term Memory Convolutional Neural Network (LSTM-CNN) architecture. This methodology benefits from the sequential data treatment capabilities of LSTM combined with the local feature extractive powers of CNN, which enable effective domain identification in Telugu texts. The model was assessed at the ICON Shared Challenge "TechDOfication 2020," scoring an F1 score of 90.01% on the validation set and 69.90% on the test set. The results indicate a great improvement over conventional models and show the tremendous efficacy of multichannel deep learning techniques for domain identification in Telugu. The proposed model will serve as a milestone toward enhancing NLP applications for regional languages while providing a scalable solution to the heightened demands for accurate thematic classification of techno-domain risks.

Keywords: Natural Language Processing (NLP), Multichannel LSTMCNN, Long Short-Term Memory (LSTM), Text Summarization, Multilingual Text Processing, F1 Score

Introduction

With the increase of textual data in many languages, several applications such as natural language processing (NLP) and information retrieval have also improved significantly in the last few years. In particular, automatic text-classification, and to large extent domain identification, is pivotal in order to successfully bring machine translation, summarization or sentiment analysis, and emotion recognition into practical use. While English language remains dominant in most fields of NLP research, there are specific challenges faced by speakers of regional languages like Telugu which is a Dravidian language, with millions of speakers around the globe. Domain identification approaches in the past were mainly reliant on statistical models such as Naive Bayes, K-Nearest Neighbors (KNN) but continue to be appropriate for addressing many situations however they are quite often unable to address the contextual and sequential details typically needed for Indian languages. Integrating deep learning techniques including recurrent and convolutional neural networks, will help to improve classification accuracy by extracting local features and capturing sequential dependencies.

Even with significant improvements in the language, the complexity of Telugu's morphology, the ambiguities of the language script, and the absence of Telugu language resources have always made the domain classification challenging. Existing text classifiers, which were largely designed for other languages, do not function very well when used for Telugu. This issue is

particularly acute in the fields of study with technical vocabulary. Therefore it is very important to use a robust and an accurate method for domain identification that will help in the improvement of the NLP applications in the multiple domains of the Telugu language.

Technical domain identification refers to the automatic recognition and categorization of a collection of unlabelled text documents into appropriate categories from a predefined list of domain classifications. The domain category set consists of six names: biochemistry, computer science, management, physics, and others. You may think of test text data as a system query; these categories are a library of documents. Machine translation, summarization, and question-answering represent but a subset of the numerous uses of domain identification. This technique constitutes the initial stage for the majority of subsequent applications.

Machine translation can subsequently deploy its resources effectively upon identifying the domain of the textual material. Research on text classification and domain identification has predominantly utilized the English language. The influence on regional languages, especially Indian languages, has garnered insufficient attention. The Dravidian language family encompasses Telugu, one of the oldest languages spoken in India.

According to the Ethnologue, Telugu boasts 93 million native speakers, positioning it as the sixteenth most spoken language in the world.Research and models may inadequately address Telugu text categorization, particularly in fields with specialist vocabulary. A novel way for tackling these difficulties using cutting-edge methodologies involves employing contemporary designs such as many channels. LSTM-CNN.

The work uses a multichannel LSTM-CNN architecture to investigate and improve Telugu Technical Domain Identification. The suggested method successfully categorizes Telugu text into pertinent technical areas by combining CNN for local feature extraction and LSTM for sequential dependency modelling. The model's effectiveness over traditional methods was demonstrated in the ICON Shared Challenge "TechDOfication 2020" evaluation, where it achieved an F1 score of 90.01% on the validation set and 69.90% on the test set. By tackling the difficulties caused by Telugu's linguistic complexity and scarce resources, this study helps to enhance Natural Language Processing (NLP) applications for the language. The value your perceptive remarks and eagerly await more conversations.

Literature Review

Murthy *et al.* (2013) performed initial research on Telugu text classification utilizing the Naive Bayes algorithm in the context of news articles. Their methodology illustrated the relevance of probabilistic models in the context of regional language processing. Swamy *et al.* (2014) employed K-Nearest Neighbors (KNN), Naive Bayes, and decision tree classifiers, demonstrating that traditional machine learning methods can effectively perform domain-level categorization for Indian language texts.

Narala *et al.* (2017) developed classification approaches that are both language-dependent and independent, highlighting the significance of integrating Telugu-specific linguistic features. Durga and Govardhan (2011) introduced an ontology-based classification method employing word frequency analysis to elucidate semantic relationships within Telugu literature.

Liu *et al.* (2020) presented an attention-based multichannel CNN model, illustrating the effectiveness of integrating local features from CNN with contextual dependencies acquired through BiLSTM. The architecture significantly impacted the multichannel hybrid design examined in this study.

Gundapu and Mamidi (2021) introduced a multichannel LSTM-CNN designed for Telugu domain identification. Their work demonstrated superior performance through the effective integration of sequential modeling and feature extraction, surpassing numerous single-channel methods. Settineni *et al.* (2023) validated the efficacy of hybrid deep learning methods, concluding that combinations of BiLSTM and CNN-LSTM surpassed standalone models in the classification of Telugu news texts.

de Lope and Graña (2023) examined developments in speech emotion recognition, thereby endorsing feature extraction methods relevant to textual emotion and sentiment indicators in Indian languages. Boddu and Reddy (2023) introduced a heuristic RNN framework aimed at automatic Telugu text categorization, specifically from handwritten sources, demonstrating the applicability of deep learning in contexts beyond typed text.

Dey *et al.* (2022) provided a comprehensive overview of Indian spoken language recognition through a machine learning lens, particularly pertinent to Telugu given its phonological complexity. Prasad and Reddy (2019) utilized multichannel CNN-LSTM models for sentiment analysis, highlighting their efficacy in extracting local and sequential features, a method incorporated in our proposed architecture.

Harish and Rangan (2020) performed a survey on the processing of Indian regional languages, highlighting significant challenges including script diversity, morphological complexity, and a scarcity of linguistic resources. Shah *et al.* (2020) investigated opinion mining for bilingual content through traditional and neural methods, tackling complexities akin to those found in Telugu NLP tasks.

Chauhan *et al.* (2024) introduced HCR-Net, a hybrid CNN-RNN model designed for handwritten character recognition, demonstrating the efficacy of hybrid deep learning approaches in character-level tasks.

Mandal *et al.* (2025) investigated the function of attention mechanisms in language identification tasks and discovered that, although they can improve performance, they are not always necessary. This

discovery is consistent with our own findings, which indicate that the integration of self-attention enhances classification in Telugu domain identification, but may also result in an additional computational burden.

Aravinda Reddy *et al.* (2019) concentrated on the use of machine learning models to detect paraphrases in Telugu. The broader challenges in Telugu NLP, particularly in capturing semantic similarity and syntactic variance, are reinforced by their work, despite the fact that it does not directly address domain identification. Our model aims to address these challenges by utilizing deep learning with attention mechanisms.

Ashikur Rahman *et al.* (2022) conducted a review of two decades of Bengali handwritten digit recognition, providing valuable comparisons for processing in low-resource languages.

Slam *et al.* (2023) examined advancements in low-resource speech recognition technologies, thereby elucidating the technical constraints associated with Telugu as a low-resource language. Khurana *et al.* (2023) assessed CNN models for Indic speech datasets, illustrating trends in the shift from classical to contemporary classification methods.

Sherstinsky (2020) conducted a comprehensive analysis of LSTM networks, highlighting their effectiveness in handling sequential data, which is a fundamental component of our methodology. Harish and Rangan (2020) identified prevalent challenges in Indian languages, whereas Dey *et al.* (2022) underscored the significance of machine learning in enhancing speech recognition for low-resource languages.

Chauhan *et al.* (2024) and Mandal *et al.* (2025) examined hybrid architectures and attention mechanisms in language-related tasks, both of which correspond to the structure of our model. Gundapu and Mamidi reaffirmed the benefits of combining CNN and LSTM for Telugu domain classification. However, our proposed model enhances this approach by incorporating self-attention, resulting in improved accuracy.

Materials and Methods

Dataset Description

Datasets used, especially those obtained via the TechDOfication 2020 Shared Task, have been thoroughly described. Included in this are the training set (68,865 documents), validation set (5,920 documents), test set (2,611 documents), and domain label dispersion (Table 1).

Table 1: Data Set Statistics

No.	Labels	Training Data	Validation Data
1	CSE	24,937	2,175
2	Phy	16,839	1,666
3	Com-Tech	11,626	977
4	Bio-Tech	7,468	588
5	Mangt	2,347	166
6	Others	5,648	392
Total		68,865	5,964

Figure 1 illustrates the distribution or aggregation of texts within the collection via visualization, indicating the quantity of documents per category or other pertinent classifications.

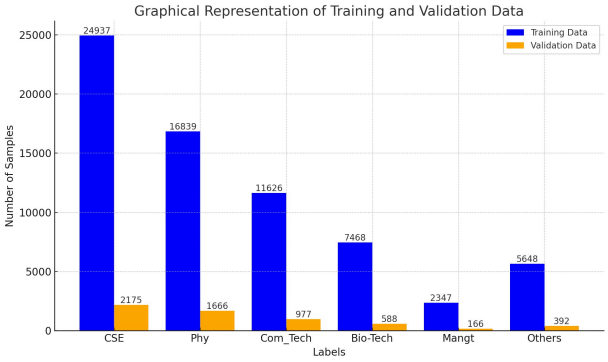

Fig. 1: Quantity of Class Samples

Preprocessing Techniques

Detail the preparation procedures, including text normalization, elimination of stop words, punctuation removal, tokenization, and the use of FastText embeddings tailored for Telugu.

Model Architecture

The specifics of the hybrid deep learning architecture that integrates BiLSTM (incorporating self-attention) and CNN components. A multichannel architecture was used to extract sequential and spatial information from the text.

Training Strategy and Parameters

Presented the hyperparameter tuning approach, including activation functions (ReLU, tanh, sigmoid), modifications to the learning rate, and optimization techniques. The training and assessment used conventional measures such as Accuracy, Precision, Recall, and F1 Score.

Implementation Tools

The model was created using Python with TensorFlow/Keras frameworks, implemented on high-performance computer infrastructure appropriate for deep learning.

When building machine learning models, the biggest subset is called the training set. To optimize the model's performance, the hyper parameters are fine-tuned using the affirmation set. The generalizability of trained models can be evaluated using the verification set, which is an independent dataset.This organized method is useful for showing how the dataset is made up, how it is divided into test, validation, and training sets, and how each subset is used while doing machine learning (Boddu & Reddy, 2023).

Figure 2 illustrates the comprehensive architecture of the proposed hybrid Multichannel LSTM-CNN model

for Telugu text classification. The pipeline begins with the input of raw Telugu text and progresses through the following stages to produce a domain prediction.

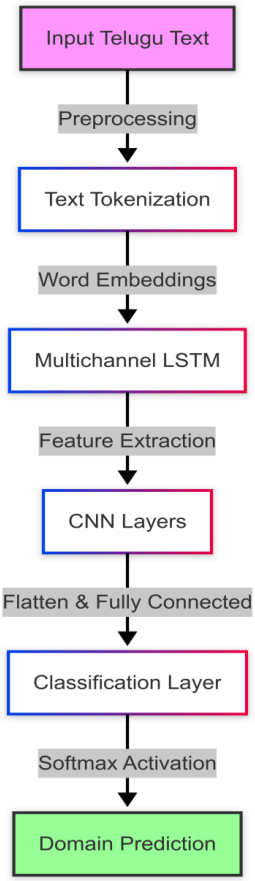

Fig. 2: Block Diagram of Multichannel LSTM-CNN

The process initiates with text preprocessing, where the raw input is cleaned and normalized through lowercasing, the removal of stop words/punctuation, and spell-checking. The cleaned text is then passed through tokenization, segmenting it into individual tokens (words or subwords) to create a structured format for the model.

Subsequently, these tokens are converted into a numerical representation via word embedding using pre-trained models like Word2Vec or FastText, which captures the semantic relationships between words.

The core of the model consists of two parallel feature extraction pathways that process the embedded sequence.

A Multichannel Bidirectional LSTM (BiLSTM) with Self-Attention pathway that captures long-range, bidirectional sequential dependencies within the text. The self-attention mechanism helps the model weigh the importance of different words for the final classification (Dey et al., 2022).

A Multi-Kernel 1D Convolutional Neural Network (CNN) pathway which operates in parallel to the LSTM, using multiple filter sizes to detect local spatial patterns and salient n-gram features within the token sequence.

The high-level features extracted from both the LSTM and CNN pathways are then aggregated and flattened into a unified feature vector. This vector is passed through a fully connected (dense) layer to perform non-linear transformation and prepare the features for the final decision.

The process concludes with the classification layer, which uses a softmax activation function to convert the outputs into a probability distribution over the potential domain categories. The category with the highest probability is selected as the final domain prediction.

This hybrid approach synergistically leverages the LSTM's proficiency in modeling long-term contextual relationships and the CNN's strength in identifying local, discriminative features, thereby enhancing overall classification performance for Telugu text.

Algorithm for Multichannel LSTM-CNN Model

Figure 3 delineates the algorithmic flow of the proposed model, illustrating the sequential processing of phrase matrices via BiLSTM and CNN layers, culminating in feature aggregation and prediction.

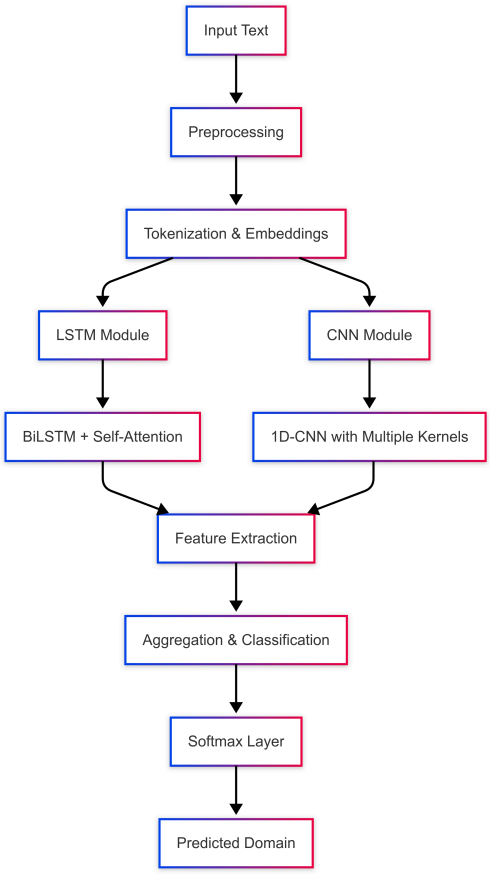

Fig. 3: MLSTM-CNN Algorithm

Figure 4 depicts the architecture of the proposed Multichannel LSTM-CNN model, in which input text is concurrently processed via distinct LSTM and CNN pathways. This dual-channel architecture allows the

model to identify both sequential patterns and local characteristics, hence enhancing classification efficacy.

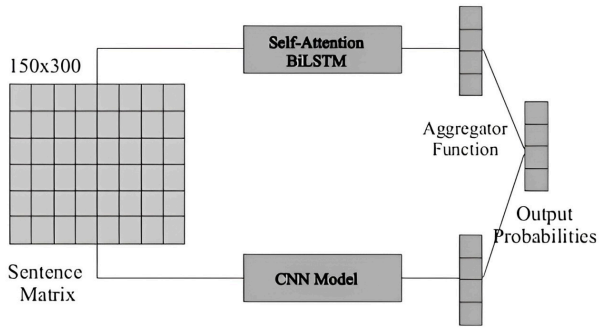

Fig. 4: LSTM-CNN Model for multiple channels

The BiLSTM network's self-attention mechanism is shown in Figure 5. To improve domain classification accuracy, this approach uses weights to identify which subword tokens are most relevant to the input context.

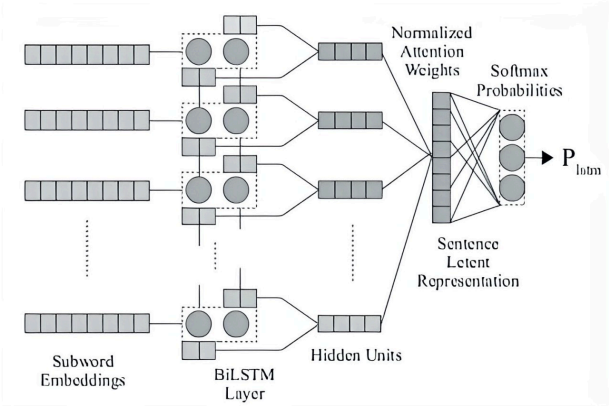

Fig. 5: Self-Attention BiLSTM

The first component of the design is a bi-LSTM classifier that uses self-attention to extract sentiment and semantic information from the text input data, they used this BiLSTM model that is based on self-attention.

As an intra-attention process called self-attention, the weights of each sub-word in the phrase are determined using a soft-max function (Prasad & Reddy, 2019).

The self-attention method takes sub-word embedding's that have been pre-trained as input and is based on the Bi-LSTM architecture. Let us assume that the sub words provide the input expression $S(W_1, W_2, ..., W_n)$.

Sketch the concealed state in reverse at its current location.

Let $\overrightarrow{h}$ represent the forward hidden state and the backward hidden state at a specific place in the BiLSTM. Ki is the outcome of integrating the forward and backward hidden states. The integration of forward and backward hidden units produces the amalgamated representations $(k_1 k_2, ..., k_n)$ (Harish & Rangan, 2020).

$$k_i = \left[\overrightarrow{h_i}; \overleftarrow{h_i}\right] \quad (1)$$

Each subword *i* in the phrase *S* is allocated a score *ei* according to the self-attention mechanism, as shown by the equation below.

$$e_i = k_i^T k_n \quad (2)$$

To create attention weight a_i, the attention score e_i is normalized.

$$a_i = \frac{\exp(e_i)}{\sum_{j=1}^{n} \exp(e_j)} \quad (3)$$

To determine the latent representation vector (i>h) of the sentence *S*, apply the equation provided below.

$$h = \sum_{i=1}^{a_i \times k_i} \quad (4)$$

CNN constitutes the second component, considering both the sequence of words in the phrase and the contextual usage of each word. To generate the requisite embedding's, use the 1D-CNN to process sentences that include Telugu fast text sub word embeddings illustrated in Figure 6.

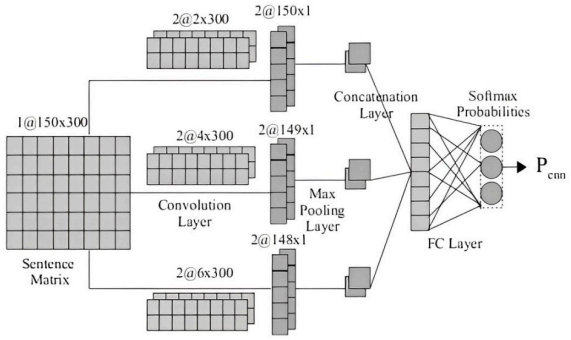

Fig. 6: CNN Classifier Model

A dS phrase embedding matrix should be first supplied to the convolution layer. Every word's d-dimensional fast ext-sub word embedding vector is represented by a row in S, which reflects the sentence length. The convolution layer does convolution operations using three distinct kernel sizes: 2, 4, and 6. various kernel sizes were employed to capture contexts of differing durations and to extract localized information for each word. The appropriate max-pooling layer was instructed to receive the output of the convolution layer (Chawla *et al.*, 2002). The maximum pooling layer preserves the word order and highlights the important parts of the feature map. Swap out the CNN initial max-pooling layer combined with a k-max-pooling layer that preserves the sequential arrangement of words to maintain the original order of the supplied phrases. The output generated by the maximum pooling layer is concatenated and fed into a fully connected layer, which is subsequently followed by a softmax layer to compute the softmax probabilities (Pcnn). The final probabilities, referred to as Pfinal, are determined through an element-wise product that averages the

softmax probabilities obtained from the CNN and BiLSTM models. Multiple aggregation methods were employed, including average, maximum, minimum, element-wise addition, and element-wise multiplication, to integrate the probabilities derived from the LSTM and CNN models. However, a product derived from elemental composition outperformed competing technologies (Harish & Rangan, 2020).

By combining sequential and local feature extraction, the Self-Attention BiLSTM-CNN model achieves improved Telugu domain identification accuracy.

Fig. 7 displays the normalized confusion matrix, illustrating the classification performance across various domains. Elevated values along the diagonal signify precise predictions, whereas off-diagonal entries illustrate confusion among similar categories, exemplified by Computer Science and Physics.

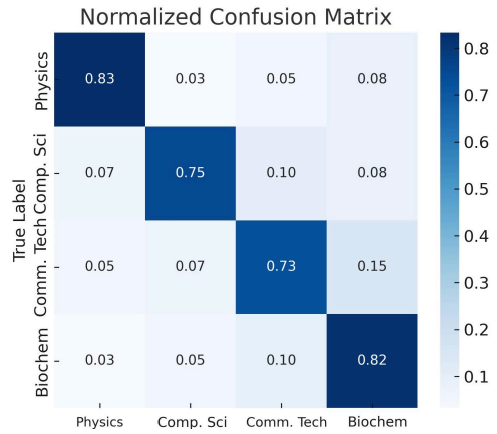

Fig. 7: Confusion Matrix Heatmap (Normalized)

The performance of the classification model across several domains is graphically represented by the normalized confusion matrix heatmap. The anticipated labels are shown in each column, while the actual labels are shown in each row. Each cell shows the percentage of predictions for a class in relation to the total instances of that class because the numbers have been normalized. While off-diagonal values indicate misclassifications, higher values near the diagonal indicate accurate classifications. For instance, a significant level of confusion between computer science and physics may indicate similar linguistic patterns or overlapping vocabulary. By emphasizing both areas that require improvement and strong classifications, this heatmap aids in diagnosing model performance.

Results and Discussion

In the beginning, we conducted our machine learning algorithm studies using various TF-IDF feature vector types. On the affirmation data set, SVM and MLP did quite well, but on the management and biochemistry data sets, they did very poorly. Furthermore, it seemed like the CNN approach that used word-of-speed merging was confused by the differentiation between computer science engineering and technology-related data points. The resemblance between the two pieces of data at the syntactic level can be the root of the confusion (Shah *et al.*, 2020).

Though it underperforms the CNN approach on data points pertaining to biochemistry and management, the self-observation-based BiLSTM method outperforms itAbout information relevant to IT, computer science, and physics. Looking at the training set, we can see that 75% of the data samples are from physics, 25% from computer science, and 25% from other domain labels. This led to the belief that an uneven training set was misclassifying biochemistry and management data. The SMOTE method, proposed by Chawla *et al.* (2002), was a failed attempt to rectify the data-skewing problem.

Table 2 compares the performance of models on the validation dataset. The Multichannel LSTM-CNN model demonstrates superior performance, achieving the highest accuracy and F1-score of 0.90, surpassing SVM, CNN, and BiLSTM models.

Table 2: Results comparison between several models (Validation Data)

Model	Accuracy	Prediction	Recall	F1-Score
SVM	0.87	0.86	0.86	0.87
CNN	0.88	0.89	0.89	0.89
BLSTM	0.87	0.88	0.87	0.88
Multi-Channel	0.90	0.90	0.90	0.90
Organizers System	-		-	-

Looking at how well the CNN and BiLSTM models worked alone, you can combine them to get better results. The constructed multiple-channel LSTM CNN method achieved better recalls and weighted F1 scores than any of our previous models (Table 3), with values of 0.90 and 0.698 on the development dataset and 0.699 and 0.699, respectively.

Table 3: Results comparison between several models (Test Data)

Model	Accuracy	Prediction	Recall	F1-Score
SVM	-	-	-	-
CNN	0.65	0.65	0.70	0.66
BLSTM	0.65	0.65	0.70	0.66
Multi-Channel	0.70	0.70	0.72	0.70
Organizers System	0.69	0.70	0.73-	0.70-

Table 4 illustrates that the Multichannel LSTM-CNN model exhibits stable performance metrics, including precision, recall, F1-score, and accuracy, on both validation and test datasets. A minor decrease in performance on the test set suggests a potential issue with overfitting.

Table 4: Evaluation of the system across many channels

Measures	Validation Data	Test Data
Precision	0.9	0.72
Recall	0.9	0.69
F1 Score	0.9	0.70
Accuracy	0.9005	0.69

According to Gundapu & Mamidi (2021) tuning is one of the most important ways to get the most out of neural network models. The activation functions of the design in particular significantly affect the final product's accuracy and learning performance. Adding non-linear activation functions to CNN layers, like ReLU (Rectified Linear Unit) and variations like Leaky ReLU, is a common way to help the network learn complicated patterns. On the other hand, long-term short-term memory (LSTM) layers usually use activation functions like sigmoid and tanh to manage how information flows through them. The article includes a picture illustrating how the Multichannel LSTM-CNN method achieved a validation set F1 score of 90.01% by meticulously the activation functions that are used, and modifying these hyper parameters. This method combines the strengths of LSTM for handling sequential dependencies with CNN for extracting local features in (Shah *et al.*, 2020) Tuning may have involved experimenting with different setups and parameters to determine the optimal performance of the model for the given task.

Figure 8 illustrates the accuracy trends of several model settings. The suggested multichannel LSTM-CNN strategy attains the best accuracy, illustrating the efficacy of integrating sequential and local feature extraction approaches.

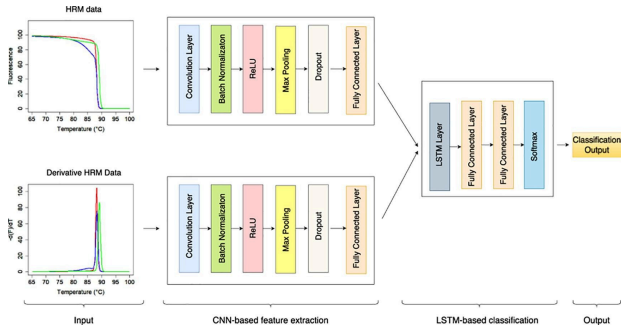

Fig. 8: Graphs for approach and accuracy

The model's performance on the validation and test datasets are compared in Figure 9. The validation metrics are still good, but there seems to have been a little dip in test performance, which might be due to overfitting and means that better generalization methods are needed.

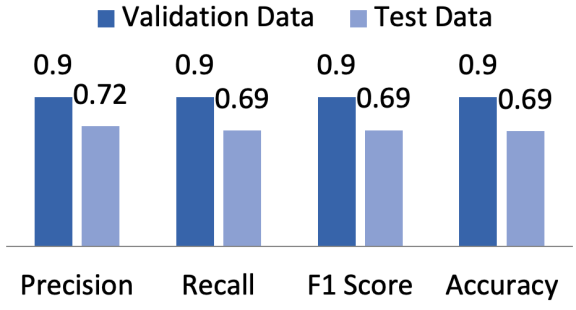

Fig. 9: Multichannel LSTM-CNN model on validation and test datasets

The Multichannel LSTM-CNN model's performance on test and validation datasets is contrasted in the graph. While the test data indicates a performance decline, with accuracy and F1-score around 0.69, suggesting some over fitting, the validation data continually displays good metrics (precision, recall, F1-score, and accuracy) (~0.9).

Figure 10 presents a comparison of the performance between the Multichannel LSTM-CNN model and a conventional CNN. The multichannel model demonstrates superior performance compared to the CNN across accuracy, precision, recall, and F1-score, underscoring the advantages of integrating sequential and local feature learning.

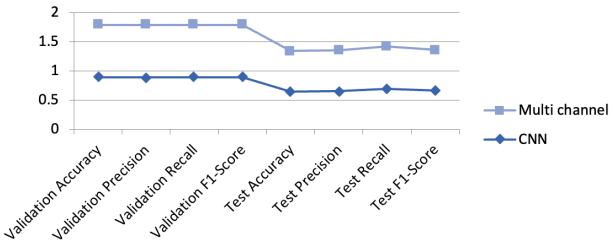

Fig. 10: Performance Comparison of Multichannel LSTM-CNN and CNN Models

The Multichannel LSTM-CNN and CNN models are contrasted in the graph using test and validation metrics. In both datasets, the Multichannel model consistently performs better than CNN, exhibiting superior accuracy, precision, recall, and F1-score.

Figure 11 presents a comparison between the Multichannel LSTM-CNN model and the BiLSTM model. The findings indicate that the multichannel strategy attains superior accuracy and F1-score, highlighting enhanced performance through the integration of CNN and BiLSTM elements.

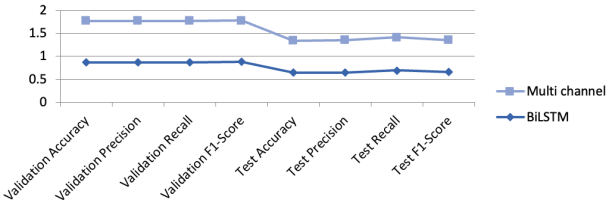

Fig. 11: Performance Comparison of Multichannel LSTM-CNN and BiLSTM Models

The Multichannel model consistently outperforms BiLSTM across all criteria, according to the graph that compares the performance of the Multichannel LSTM-CNN and BiLSTM models. Higher accuracy, precision, recall, and F1-score are attained by the Multichannel model on both test and validation datasets.

Table 5 presents a comparison of the validation performance among CNN, BiLSTM, and the Multichannel LSTM-CNN. The multichannel model consistently demonstrates superior metrics, thereby validating the efficacy of the hybrid approach.

Table 5: Performance Comparison (Validation Data)

Metric	Multi-Channel	CNN	BiLSTM
Accuracy	0.90	0.88	0.87
Precision	0.90	0.88	0.87
Recall	0.90	0.88	0.87
F1-Score	0.90	0.89	0.87

Table 6 presents the performance of the model on the test data. The Multichannel LSTM-CNN demonstrates superior performance compared to alternative models, attaining the highest accuracy and F1-score, thereby validating its generalization capability despite a slight decline from validation outcomes.

Table 6: Performance Comparison (Test Data)

Metric	Multi-Channel	CNN	BiLSTM
Accuracy	0.69	0.64	0.64
Precision	0.69	0.65	0.65
Recall	0.72	0.69	0.69
F1-Score	0.69	0.66	0.65

Table 7 evaluates the suggested model in comparison to previous studies in Telugu text categorization. The Multichannel LSTM-CNN attains the highest documented F1-score (90.01%), exceeding prior deep learning and statistical methodologies.

Table 7: Table Comparison of Proposed Model with Previous Studies

Study	Method	Dataset	Accuracy (%)	F1-Score (%)
Murthy (2013)	Naive Bayes	Telugu News (800 samples)	72	70
Swamy *et al*. (2014)	KNN, Naive Bayes, Decision Trees	Telugu Articles	75	72
Narala *et al*. (2017)	Language-dependent models	Telugu Texts	78	75
Durga & Govardhan (2011)	Word Frequency Ontology	Telugu Literature	76	73
Liu *et al*. (2020)	Attention-based CNN	Telugu Corpus	84	82
Gundapu & Mamidi (2021)	Multichannel LSTM-CNN	Technical Telugu Corpus	88	85
Proposed Method (2025)	Self-Attention BiLSTM + CNN	TechDOfication 2020 (76K docs)	90.05	90.01

Performance Comparison

Figure 12 provides a consolidated comparison of CNN, BiLSTM, and Multichannel LSTM-CNN models. The Multichannel LSTM-CNN consistently achieves the best performance across all evaluation metrics, confirming its effectiveness for Telugu domain classification.

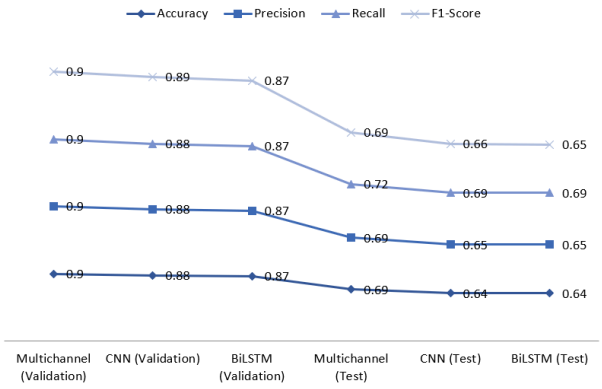

Fig. 12: Performance Comparison of Multichannel, CNN, and BiLSTM Models

A comparison of the Multichannel LSTM-CNN, CNN, and BiLSTM models' performances on test and validation datasets is shown in the table. The Multichannel LSTM-CNN regularly beats CNN and BiLSTM in every category, demonstrating superior generalization on the test dataset. It also consistently earns the greatest results across all metrics (Accuracy, Precision, Recall, and F1-Score).

Although the suggested Multichannel LSTM-CNN model for Telugu domain identification performs well, it has certain drawbacks in other domains:

It is challenging to categorize underrepresented fields like biochemistry and management because of the dataset's strong bias towards the domains of physics, computer science, and communication technology. The model found it difficult to increase classification accuracy for underrepresented categories, despite efforts to balance the dataset using SMOTE.

Because of their similar lexicon, technical fields like computer science and communication technology were hard for the CNN model to differentiate. When terminology and language structure are shared across several domains, misclassification mistakes happen.

On validation data, the model's F1-score was 90.01%; however, on test data, it fell to 69.90%, suggesting that over fitting may have occurred. This implies that the model may function well on training-set-like data but poorly on test data that hasn't been seen yet.

Managing Telugu's Complex Morphology and Syntax as Telugu is a morphologically rich language; the model might have trouble with words that have several context-dependent interpretations. Errors may be introduced by some preprocessing methods, such as word embeddings and Telugu translations from Hindi and English.

The model works well in areas with organized vocabulary, but it might have trouble in emerging or highly specialized technical disciplines where the training data does not adequately represent the language.

The results confirm the efficacy of the Multichannel LSTM-CNN method in capturing sequential dependencies and local features in Telugu texts. This hybrid strategy offers a promising approach for

enhancing domain classification in low-resource, morphologically rich languages. Future research will focus on improving domain generalization, optimizing dataset balance, and investigating advanced deep learning architectures, including transformers.

Several key improvements are planned to enhance model performance further. To address the issue of training sample imbalance, we will implement advanced data augmentation techniques specific to the Telugu language. Furthermore, feature representation will be strengthened by employing domain-specific embeddings trained on larger, relevant corpora. Finally, we plan to explore more complex hybrid model architectures that can incorporate additional contextual metadata, thereby providing a richer foundation for classification.

Funding Statement

No funding was received for this manuscript.

Conclusion

In this study, developed a multichannel strategy that combines CNN and LSTM benefits. This paradigm expresses mood, local and global dependencies, and both all in one declaration. We beat supervised ML methods, standalone LSTM, and CNN on the Telugu TechDOfication dataset.

As indicated earlier, we will successfully resolve the problem of skewed data in subsequent research. may further augment the efficiency of clear-cut situations in the fields of information technology and computer science.

Acknowledgment

The authors extend their heartfelt appreciation to the organizers of the TechDOfication 2020 Shared Task for supplying the dataset used in this research. We express our gratitude to our respective institutions, Sathyabama Institute of Science and Technology, Aditya University, Vignan's Institute of Information Technology, Vasavi College of Engineering, and Vignan's Institute of Engineering for Women, for their assistance and encouragement throughout this research endeavor.

Ethics

This research was performed in compliance with established ethical standards. All utilized data were publicly accessible and did not include any personal, sensitive, or confidential information. No human participants were involved; thus, ethical approval and informed consent were not necessary. The study adheres to the ethical standards set forth by the institution and the journal.

Funding Information

Members of TechDOfication 2020's Task-h provided funding for this project.

References

Aravinda Reddy, D., Anand Kumar, M., & Soman, K. P. (2019). Paraphrase Identification in Telugu Using Machine Learning. *Advances in Big Data and Cloud Computing*, *750*, 499—508. https://doi.org/10.1007/978-981-13-1882-5_43

Ashikur Rahman, A. B. M., Hasan, Md. B., Ahmed, S., Ahmed, T., Ashmafee, Md. H., Kabir, M. R., & Kabir, Md. H. (2022). Two Decades of Bengali Handwritten Digit Recognition: A Survey. *IEEE Access*, *10*, 92597–92632. https://doi.org/10.1109/access.2022.3202893

Boddu, R., & Reddy, E. S. (2023). Novel Heuristic Recurrent Neural Network Framework to Handle Automatic Telugu Text Categorization from Handwritten Text Image. *International Journal on Recent and Innovation Trends in Computing and Communication*, *11*(4s), 296–305. https://doi.org/10.17762/ijritcc.v11i4s.6567

Chauhan, V. K., Singh, S., & Sharma, A. (2024). HCR-Net: A Deep Learning Based Script Independent Handwritten Character Recognition Network. *Multimedia Tools and Applications*, *83*(32), 78433–78467. https://doi.org/10.1007/s11042-024-18655-5

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, *16*, 321–357. https://doi.org/10.1613/jair.953

de Lope, J., & Graña, M. (2023). An Ongoing Review of Speech Emotion Recognition. *Neurocomputing*, *528*, 1–11. https://doi.org/10.1016/j.neucom.2023.01.002

Dey, S., Sahidullah, M., & Saha, G. (2022). An Overview of Indian Spoken Language Recognition from Machine Learning Perspective. *ACM Transactions on Asian and Low-Resource Language Information Processing*, *21*(6), 1–45. https://doi.org/10.1145/3523179

Durga, K., & Govardhan, A. (2011). Ontology Based Text Categorization - Telugu Documents. *International Journal of Scientific & Engineering Research*, *2*.

Gundapu, S., & Mamidi, R. (2021). *Multichannel LSTM-CNN for Telugu Technical Domain Identification*. https://doi.org/10.48550/arXiv.2102.12179

Harish, B. S., & Rangan, R. K. (2020). A Comprehensive Survey on Indian Regional Language Processing. *SN Applied Sciences*, *2*(7), 1204. https://doi.org/10.1007/s42452-020-2983-x

Khurana, S., Dev, A., & Bansal, P. (2023). SER: Performance Evaluation of CNN Model Along with an Overview of Available Indic Speech Datasets, and Transition of Classifiers From Traditional to Modern Era. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 85. https://doi.org/10.1145/3605778

Liu, Z., Huang, H., Lu, C., & Lyu, S. (2020). *Multichannel CNN With Attention for Text Classification*. https://doi.org/10.48550/arXiv.2006.16174

Mandal, A., Pal, S., Dutta, I., Bhattacharya, M., & Naskar, S. K. (2025). Is Attention Always needed? A Case Study on Language Identification from Speech. *Natural Language Processing*, *31*(2), 250–276. https://doi.org/10.1017/nlp.2024.22

Murthy, G. V., Vardhan, B. V., Sreenivas, M., & Reddy, P. V. (2013). Text Classification Using Text Summarization - A Case Study on Telugu Text. *International Journal of Advanced Research in Computer Science and Software Engineering*, *3*(7), 1399–1403.

Narala, G., Swapna, B., Padmaja Rani, B., & Ramakrishna, K. (2017). Telugu Text Categorization Using Language Models. *Global Journal of Computer Science and Technology*, *16*(4).

Prasad, G. S. C., & Reddy, K. A. N. (2019). Sentiment Analysis Using Multi-Channel CNN-LSTM Model. *Journal of Advanced Research in Dynamical and Control Systems*, *11*(12), 489–494. https://doi.org/10.5373/jardcs/v11sp12/20193243

Settineni, P., Veerendra, B., Jaideep Reddy, M. S., Samuel, P., & Sumathi, D. (2023). Comparative Analysis of Deep Learning Models for Telugu News Text Classification. *2023 OITS International Conference on Information Technology (OCIT)*, 847–854. https://doi.org/10.1109/ocit59427.2023.10430996

Shah, S. R., Kaushik, A., Sharma, S., & Shah, J. (2020). Opinion-Mining on Marglish and Devanagari Comments of YouTube Cookery Channels Using Parametric and Non-Parametric Learning Models. *Big Data and Cognitive Computing*, *4*(1), 3. https://doi.org/10.3390/bdcc4010003

Sherstinsky, A. (2020). Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network. *Physica D: Nonlinear Phenomena*, *404*, 132306. https://doi.org/10.1016/j.physd.2019.132306

Slam, W., Li, Y., & Urouvas, N. (2023). Frontier Research on Low-Resource Speech Recognition Technology. *Sensors*, *23*(22), 9096. https://doi.org/10.3390/s23229096

Swamy, M. N., Hanumanthappa, M., & Jyothi, N. M. (2014). Indian Language Text Representation and Categorization Using Supervised Learning Algorithm. *2014 International Conference on Intelligent Computing Applications*, 300–304. https://doi.org/10.1109/icica.2014.89